*Journal of Computer Science*

Research Article

# Enhancing Facial Expression Recognition Accuracy Through Haar Cascade-Based Feature Extraction

**Thambusamy Velmurugan[1] and Lakshminarayanan Meena[2]**

[1]*Department of Computer Science, Dwaraka Doss Goverdhan Doss Vaishnav College, Chennai, India*
[2]*Department of Computer Applications, Dwaraka Doss Goverdhan Doss Vaishnav College, Chennai, India*

*Article history*
Received: 08-01-2025
Revised: 27-09-2025
Accepted: 24-10-2025

**Corresponding Author:**
Thambusamy Velmurugan
Department of Computer Science, Dwaraka Doss Goverdhan Doss Vaishnav College, Chennai, India
Email: velmurugan_dgvc@yahoo.co.in

**Abstract:** Facial Expression Recognition (FER) has a significant interest because of its increased applications in Human-Computer Interaction (HCI), like video interactions, emotion analysis, image indexing, and retrieval and many more. In the discipline of computer vision, pattern recognition, and artificial intelligence FER has become an extremely active research topic. A crucial component of FER systems is feature extraction, which involves identifying unique features from facial images and representing them in a quantized form. This research work utilizes a robust feature extraction method namely the Haar Cascade Algorithm. This work is motivated by the drawbacks in the existing feature extracting techniques which decreases the accuracy in predicting facial expressions. The main objective of this work is to demonstrate the impact of Haar Cascade based feature extraction technique in the accuracy of predicting emotions. Previous studies are reviewed to evaluate the effectiveness of existing feature extraction methods. Based on this analysis, the most suitable approach is identified to improve accuracy in recognizing the seven basic facial expressions: happiness, sadness, anger, surprise, disgust, fear, and neutral.

**Keywords:** Feature Extraction, Facial Expression Recognition, Human-Computer Interaction, Haar Cascade Algorithm, Convolutional Neural Network

## Introduction

Facial Expression Recognition (FER) exhibits a vital role in understanding human emotions and mental states through the analysis of the facial expressions such as happiness, sadness, anger, fear, surprise, neutral and disgust (Zhao and Zhang, 2016). It implies the fact that a significant portion of about 55% of information about emotions is conveyed through facial expressions (Huang and Huang, 1997). The importance and applications of FER include Human-Computer Interaction (HCI) enabling computers to interpret and respond to human emotions, enhancing natural interaction between users and machines thereby applying in virtual assistants, gaming, and user interface design (Siddiqi *et al*., 2012). Facial expression recognition mainly uses two types of feature extraction methods: geometric-based and appearance-based techniques. Geometric-based methods focus on the facial structure and shapes of key features to identify expressions. While the appearance based feature extraction technique focuses on the texture of the face exploring the wrinkles and scratches on the face (Dubuisson *et al*., 2001). The diversity of facial expressions and their resemblance to one another are the major challenges in FER (Ilbeygi and Shah-Hosseini, 2012). The variation in wrinkles and textures is less noticeable when compared with the variations in geometric elements, hence variations in facial expressions are more noticeable in geometric components (Choudhary and Shukla, 2020). Facial feature extraction in emotion recognition systems is a critical stage that determines the system's accuracy and performance (Abouyahya *et al*., 2016).

Through selecting and implementing precise algorithms, researchers can substantially improve the ability of the system to interpret and classify emotions based on facial expressions with accuracy. (Lai and Ko, 2014). In this research work the Haar Cascade algorithm is used to detect facial features, like eyes, nose and mouth, from an input image. The CNN algorithm, which is responsible for determining the expression, is input with these extracted features. Experiments on an openly

**DOI: 10.3844/jcssp.2025.2917.2927**

available data set illustrated that the proposed technique yields 97.14% accuracy. This suggests that the technique performs very effectively for accurately determining facial expressions.

### *Literature Survey*

Numerous research carried out by many people in the globe imply the importance of FER in the HCI field. Some of the research papers on different feature extraction techniques used in FER model and the performance of the models are compared in Table 1.

Kommineni *et al*. (2021) introduced a hybrid algorithm that combines the Dual-Tree M-Band Wavelet Transform (DTMBWT) technique with the Gray-Level Co-occurrence Matrix (GLCM) method. The performance of this algorithm is evaluated using the Japanese Female Facial Expression (JAFFE) database to identify the seven distinct facial expressions. The findings from the experiment demonstrate that this approach has the highest precision of 99.53% at the 4th decomposition level, thereby identifying facial expressions more appropriately than other methods. A new image feature extraction method is introduced based on Gabor filters in a research study done by Sadeghi and Raie (2019), in which both Gabor convolution responses and filter orientations in a histogram-based feature vector are utilized and experimented on three different datasets (including CK+, MMI, and SFEW) in both controlled and uncontrolled conditions. Concluding the proposed feature extraction method outperforms the state-of-the-art texture descriptors.

Zhou *et al*. (2006) proposed a method based on selective feature extraction to extract features in order to resolve the problem of increased redundant information and the difficulty in obtaining the templates. Experiment results shows that the method can obtain satisfied recognition rate and has strong robustness to illumination variety. Gupta *et al*. (2011) proposed a hybrid method for feature extraction that combines DCT, Gabor Filter, Wavelet Transform, and Gaussian distribution. The proposed techniques have a recognition rate of 93.4% while facial expression recognition using individual methods namely DCT method, Wavelet Transform method, Gabor Filter method, Gaussian Distribution method presented a recognition rates as follows 75.94, 64.11, 67.5 and 63.14 % respectively.

The features are extracted from active facial patches based on the position of facial landmark points in a study by Happy and Routray (2015). Further the features from different sub regions are concatenated to represent the whole image. Thereby improving the performance and reducing the computational cost. The proposed method performed well for both CK+ and JAFFE databases and a promising accuracy was obtained in recognizing all expression classes of multiple subjects. Concatenation of shape and appearance features outperforms the individual features in case of expression recognition.

**Table 1:** Comparison of Feature Extraction Techniques

| S. No | Feature Extraction Method | Advantages | Disadvantages |
|---|---|---|---|
| 1 | Dual-Tree M-Band Wavelet Transform (DTMBWT) with energy, entropy, and GLCM (Kommineni *et al*., 2021) | High precision (99.53%) at $4^{th}$ decomposition level. Effective for all 7 basic facial expressions<br>Captures spatial-frequency texture features efficiently | Complex computation at high decomposition levels<br>May require fine-tuning for different datasets |
| 2 | Gabor Filter-based Histogram Method (Sadeghi and Raie, 2019) | Robust to both controlled and uncontrolled conditions<br>Outperforms state-of-the-art texture descriptors<br>Captures orientation and frequency effectively | High computational cost.<br>Sensitive to parameter selection (e.g., orientation, scale) |
| 3 | Selective Feature Extraction (Zhou *et al*., 2006) | Reduces redundant information<br>Robust against illumination variations<br>Improves recognition rate | May not generalize well to all datasets<br>Template generation still has some challenges |
| 4 | Hybrid Method (DCT + Gabor Filter + Wavelet Transform + Gaussian Distribution) (Gupta *et al*., 2011) | Improved recognition rate (93.4%) using hybrid approach<br>Combines strengths of multiple techniques | Individual methods perform poorly alone<br>High feature dimensionality<br>Increased processing time |
| 5 | Facial Landmark-based Active Patch Extraction (Happy and Routray, 2015) | Reduces computational cost<br>High accuracy on CK+ and JAFFE<br>Better representation by combining sub-regions | Performance depends on accuracy of facial landmark detection<br>May miss holistic features |
| 6 | Preprocessed Image-Based Feature Extraction for CNN Input (Kumar *et al*., 2016; Zhang *et al*., 2012; Krithika and Priya, 2021) | Useful for large-scale datasets<br>CNNs handle complex patterns and variation<br>High accuracy in classifying 7 expressions | Relies heavily on quality of preprocessing<br>Requires large annotated datasets for training |

## Materials and Methods

In this research work the techniques and additional information needed in feature extraction to achieve the highest level of accuracy in predicting facial expressions are analyzed. Facial images input to the FER model are preprocessed. The characteristic features are extracted from the resulting image after preprocessing (Kumar *et al*., 2016). Two sets of facial images are required to train and test the model. A large number of facial images are collected from publicly available sources and preprocessed to extract important features for training the facial expression recognition system. For testing, features are extracted from facial images captured directly or obtained from social media. (Zhang *et al*., 2012). The extracted facial features are utilized by the CNN model to predict the seven major facial expressions accurately (Krithika and Priya, 2021).

### *Feature Extraction*

Features are a key characteristic that describes the complete facial image. A feature is an essential component of the data generated to solve computational issues related to certain applications (Janu *et al*., 2017). It is essential to extract relevant and meaningful aspects from the facial images so that the appropriate classification methods can be implemented (Mattela and Gupta, 2018). Feature extraction can be accomplished by numerous mathematical models like image processing techniques and computational intelligence tools like neural networks or fuzzy logic. In order to identify various facial expressions, it is mandatory to recognise changes in the features of the face, including the eyes, cheeks, lips, and brows (Tsai and Chang, 2018). In general, the alterations in the form and development of a furrow or wrinkles are noticeable features that are useful for examination of the expressions on faces (Vedantham and Reddy, 2020). The feature extraction techniques are classified into four general approaches namely feature-based approach, appearance-based approach, template-based approach, and part-based approach. The classification of various feature extraction techniques is shown in Figure 1 (Mayya *et al*., 2016). Appearance-based approach aims to recognize faces using global representations, which implies that they rely on the entire facial image rather than just certain local features. These techniques are also known as holistic-based techniques, which employ the information from the whole face patch and then execute different modifications on it to derive an exact representation for facial recognition (Pei and Shan, 2019).

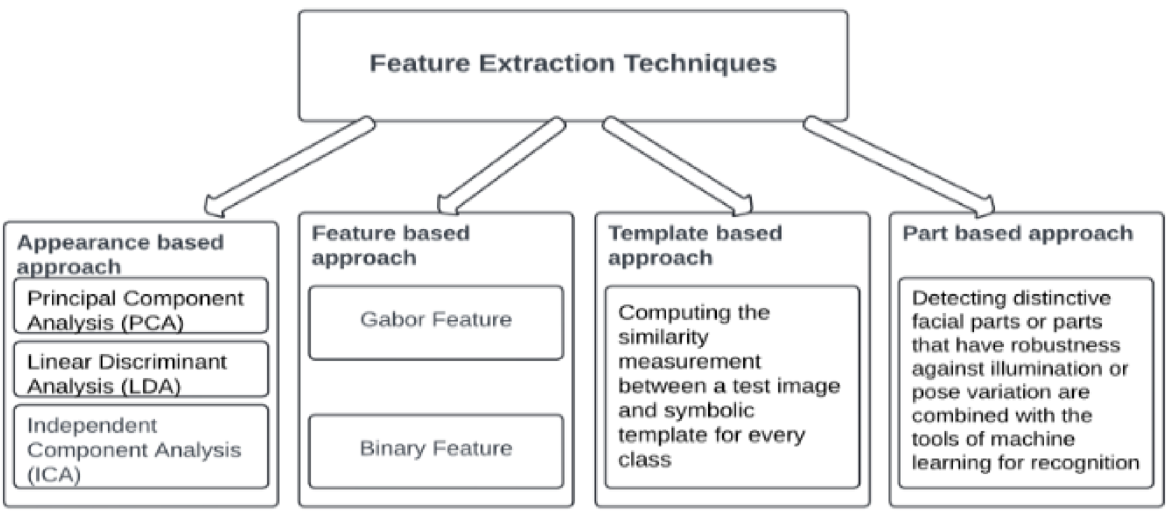


**Fig. 1:** Feature extraction for facial images

These methods transform the entire patch producing the feature vector and the foundation for these transformations are generally derived from statistics. This approach uses a number of techniques, including PCA, LDA, ICA, etc. (Tseng and Chen, 2018).

In contrast with appearance-based approaches, which focus more on statistical learning and analysis, feature based approaches employ image processing, the domain of human knowledge (Ilbeygi and Shah-Hosseini, 2012). In order to reduce the input facial image to a vector of geometric features, these approaches first process the input images to identify and extract the distinct facial features, such as the mouth, nose, eyes, and other standard facial marks (Liu *et al*., 2021). After that, the geometric relationships among all of those facial points are computed. The most popular approaches for feature extraction in facial recognition are Local Binary Pattern (LBP) and the Gabor wavelet feature (Luo *et al*., 2013).

**DOI: 10.3844/jcssp.2025.2917.2927**

Template matching in template-based approaches performs similar to utilizing a distance metric to identify a face. Initially a set of symbolic templates for every class is set and then the similarity between a test image and each class is determined. The class with the highest similarity score is selected as the correct match (Singh and Nasoz, 2020) In the part based approach for identification of faces, information derived from distinctive facial features or features that are resistive to brightness or variations in posture can be used (Kola and Samayamantula, 2021). To differentiate part based approaches from feature-based approaches; significant portions of the facial image are identified and then combined with machine learning tools for recognition (González-Lozoya *et al*., 2020). While feature-based approaches extract features from the facial feature points or from the entire face and use feature comparison to accomplish recognition. In order to overcome the challenges in facial recognition such as pose, different lighting conditions, and face size, the speed at which the facial features are extracted plays an important role (Li *et al.*, 2020). To accomplish high speed and utmost accuracy some of the above mentioned approaches are combined and utilized effectively in recognizing facial expressions (Rizwan *et al*., 2020).

### *Importance of Haar Cascade*

Haar Cascade is chosen over other feature extraction methods for several practical reasons, particularly in the context of early computer vision tasks such as face detection. Haar-like features can be computed extremely fast using integral images, allowing constant-time feature computation regardless of size. The cascade structure enables early rejection of negative regions, thereby improving detection speed. Haar-like features are simple rectangular patterns (like edges, lines, and center-surround structures) that are effective for capturing contrast differences, especially useful for tasks like face detection thereby achieving simplicity and effectiveness of the emotion detection model. These features work well with frontal faces under controlled lighting and scale.

The use of AdaBoost in training selects the most relevant Haar features from a large set, helping the classifier focus on the most discriminative parts of the image boosting for feature selection. Haar Cascade was one of the first practical and accurate methods for object detection that could be used in real-time applications like webcams and mobile devices proving its historical context with early success even before the use of deep learning and more advanced descriptors like Histogram of Oriented Gradients (HOG) or Scale-Invariant Feature Transform (SIFT). Haar Cascade was widely adopted at a time when computing resources were limited. Its low computational and memory requirements made it suitable for embedded systems and devices with limited processing power. Hence Haar Cascade is chosen over other feature extraction techniques for its speed, simplicity, and effectiveness in feature extraction in facial images.

### *Description of Dataset*

A FER system requires two types of datasets: One for system training and one for system validation or testing. A sizable dataset, which can be acquired from the available repositories, can be used for training the model (Li and Lima, 2021). The training dataset for this research work is derived from FER2013. In this work 1,750 facial images belonging to seven expression classes are categorized in the training set.

The details of the images used to test the model and extract the features are described in Table 2. Indian subjects between the ages of 10 and 50 were asked to pose with various facial expressions, and their expressions were captured. This work shows the findings from four of these subjects. The preprocessed images were used to extract facial features, including the mouth, nose, right eye, and left eye, which are used for predicting facial expressions.

**Table 2:** Details of dataset for testing the model

| Properties of the dataset | Details |
|---|---|
| Subject count | 350 |
| Age | 10 to 50 years |
| Origin region of subjects | Indian |
| Size of image | 2 KB |
| Types of basic expressions | 7 |
| Method of capturing image | Posed |

### *Proposed Feature Extraction Technique*

The Facial Expression Recognition system contains a technical feature that allows it to use hardware resources offered for free in Google Colab. The dataset can be acquired from Kaggle and preprocessed before extracting the most important characteristics for training the model with the Python library Tensorflow 2.0. The dataset for evaluating the facial expression recognition system is available by capturing facial images from real world or by downloading them from social media (Li *et al*., 2023). The most important features are extracted by implementing the Haar Cascading algorithm with Python using the OpenCV library, which is a library of programming functions for real-time computer vision. The feature extraction technique is accomplished with the Haar Cascading algorithm which identifies and extracts the most prominent features like left eye, right eye, nose and mouth. The extracted features are then input to the CNN classifier which predicts facial expressions with utmost accuracy. The following libraries are essential in Python for all Machine Learning tasks: Pandas and NumPy for data manipulation and Matplotlib for visualizing data and model performance. Figure 2 explains the proposed feature extraction technique for achieving optimized accuracy in detecting facial expressions. Facial images either captured or downloaded from social media are pre-processed which is the initial step in FER. After pre-processing features that are most important namely left eye, right eye, nose and mouth are extracted using Haar cascade algorithm (Gogić *et al*., 2020). The CNN classifier, which has been pre-trained to classify the seven facial expressions, receives the extracted features. The facial expression can be accurately predicted from the features (Nan *et al*., 2022).

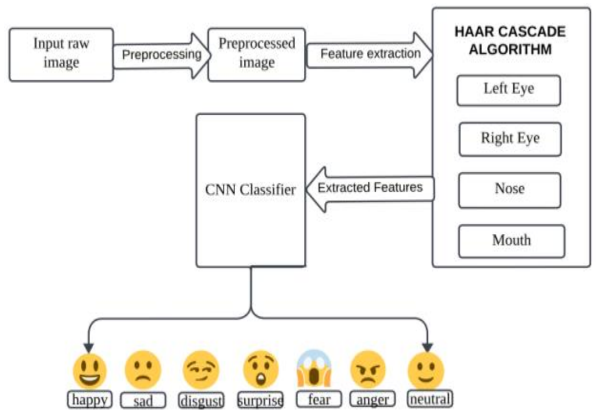


**Fig. 2:** Proposed feature extraction technique

## *Haar Cascade Algorithm*

The Haar cascade algorithm is a machine learning technique that detects objects in images, including faces, in real-time. The Haar features are extractable through the algorithm by determining the likelihood of a point to become a part of an object. The following are the steps that explain Haar Cascade algorithm:

Step 1: Import the required libraries namely cv2 for opencv, numpy for numerical computation and tensorflow for the TensorFlow object detection model

Step 2: The Haar Cascade classifier from a pre-trained xml file is loaded to detect faces from any image

Step 3: The TensorFlow object detection model is loaded thereby defining the path to the frozen inference graph, the label map and the number of classes in the model

Step 4: Firstly, the image is loaded and then converted to grayscale by the function which defines the attributes namely the image path, the TensorFlow detection graph, the category index and the minimum score threshold for detection

Step 5: Now the faces in the image are detected by the Haar Cascade classifier. Over each detected face the function iterates thereafter drawing a rectangle around it. Identifying areas that match the Haar-like features learned during training the classifier slides across the image at various scales and positions. Resulting in a bounding box around the detected facial features (eyes, nose, and mouth)

A cascade of classifiers forms the Haar Cascade algorithm. Every level of the cascade uses a set of Haar features to progressively filter out regions lacking face contours. The method detailed below consists in the tasks listed below.

## *Integral Image Calculation*

We accelerate the Haar feature computation using an integral image representation. This representation facilitates quick computation of the pixel value total inside a rectangular area. Here is defined the integral image at any point ($x$, $y$):

$$II(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j) \tag{1}$$

Where:

I ($i$, $j$) = Pixel intensity value at point ($i$, $j$)
II ($x$, $y$) = Integral image value at point ($x$, $y$)

Using four reference points from the integral image, the study may determine the sum of the pixel values inside any rectangle area $R$:

$$S(R) = II(x2, y2) - II(x1, y2) - II(x2, y1) + II(x1, y1) \tag{2}$$

Where:

$S$ ($R$) = Sum of pixel values within region $R$.
($x_1$, $y_1$) and ($x_2$, $y_2$) = Top-left and bottom-right corners of the region.

## *AdaBoost Classifier*

The AdaBoost (adaptive boosting) algorithm is used in the cascade classifier to select among a great variety of Haar features accessible the ones with the strongest degree of discrimination. Every feature in line with the degree of accuracy with which the AdaBoost algorithm labels them receives weights.

Initialize equal weights for all features.
For each feature, compute the classification error:

$$\epsilon = \sum_{i=1}^{N} w_i \, | h_i(x) - y_i | \tag{3}$$

Where:

$W_i$ = Weight for sample I
$h_i\ (x)$ = Classifier output for sample I
$y_i$ = True label for sample I

Update the weights based on the error:

$$w_i \leftarrow w_i \cdot e^{-\alpha y_i h_i(x)} \tag{4}$$

Where:
$\alpha$ = weight adjustment factor based on classification accuracy Repeat the process until the desired classification accuracy is achieved.

### *Cascade Classification*

The Haar Cascade classifier is composed of several weak classifier stages stacked in a cascade sequence. On the other hand, a window is thrown away since failing all the stages renders it not considered as a face. Every stage has the following decision function:

$$D(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^{n} h_i(x) \geq T \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Where:

$h_i(x)$ = Weak classifier output at stage $I$
$T$ = Threshold value for positive classification

The first stages of the cascade are supposed to rapidly eliminate most non-face areas, so simplifying the computation by eliminating most of its complexity. To capture different features and scale changes in an object, rectangles of different sizes and shapes are utilized. Thus, the primary advantage of Haar features is their capacity to represent three patterns.

Edge features: Depending on how we position the rectangular region, the edges may be either vertical or horizontal. They are helpful in defining the borders between various image feature sections.

Line features: In an image these patterns represent the diagonal edges. The lines and contours of an object are identified using these patterns.

Center-surrounded features: The difference in intensity between the center of a rectangular region and its surrounding area is detected by this feature. This is helpful in identifying images with a distinct shape or pattern.

The Figure 3 shows how different facial features like the eyes, nose and mouth are represented by Haar features namely the edge features and line features. To detect eyebrow, Haar edge feature is used because forehead and eyebrow form lighter pixels to darker pixel like image. Similarly, to detect lips Haar line feature is used with lighter-darker-lighter pixels. To detect nose, we might use darker-lighter Harr like feature from edge feature and so on. The features for each of the five rectangular areas are obtainable by subtracting the sum of pixels under the white region from the sum of pixels under the black region. Therefore, given these five rectangular regions and their corresponding difference of sums, we can form features that can classify parts of a face. Then the important features extracted from the preprocessed facial image is fed to a CNN, which recognizes the state of mind of a person, giving utmost accuracy in predicting the facial expressions (Meena and Velmurugan, 2023). A CNN processes data by using layers of filters that automatically learn features from input images. The core operation is the convolution, defined as:

$$Y(i,j) = (X * K)(i,j) = \sum_{m=-k}^{k} \sum_{n=-k}^{k} X(i+m, j+n) \cdot K(m,n) \tag{6}$$

Where:

- $X$: Input feature map
- $K$: Convolution kernel (filter)
- $Y$: Output feature map
- $i,j$: Spatial indices

After convolution, an activation function (like ReLU: $f(x) = max(0,x)$) introduces non-linearity. The pooling operation (e.g., max pooling) down-samples feature maps. The final feature maps are flattened and passed through Fully Connected (FC) layers for classification. CNNs are trained using cross-entropy loss:

$$L_{CE} = -\sum_{i=1}^{C} y_i \cdot \log(\hat{y}_i) \tag{7}$$

Where $y$ is the true label and $\hat{y}_i$ is the predicted probability for class $i$.

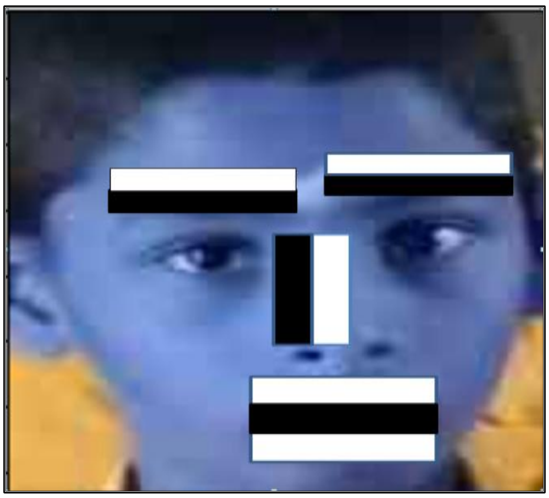

**Fig. 3:** Facial features represented by Haar features

**DOI: 10.3844/jcssp.2025.2917.2927**

## Results and Discussion

In this study, the CNN model and the Haar cascade algorithm for feature extraction are implemented using Python libraries like OpenCV and TensorFlow on GPU NVIDIA version 375.74 from nvidia-375. This work focuses on building a CNN to classify grayscale facial images into one of seven emotion categories. The system uses data preprocessing, data augmentation, a deep learning model, and evaluation metrics to assess performance. Data Augmentation is done to improve model robustness and prevent overfitting, the augmentations applied using Keras' Image Data Generator are as follows width shift range: ±10%, height shift range: ±10%, horizontal flipping, rescaling pixel values from 0–255 to 0–1, validation split: 20% for validation. A CNN built using the Keras Sequential API consists of multiple convolutional, normalization, pooling, and dense layers. The main activation function after every convolutional layer and dense layer (except the final one) is the Rectified Linear Unit (ReLU). Categorical Crossentropy is used to implement loss function is suitable for multi-class problems. Adam optimizer is used to optimize with a learning rate of 0.0001. Metrics used to evaluate is accuracy of prediction of emotions. Training Setup is done with Callback techniques like Early Stopping, that monitors validation accuracy with a patience of 15 epochs which avoids overfitting and Model Checkpoint, which saves the best model weights on maximum validation accuracy. In training configuration, the epochs are set to 200 (with early stopping) and batch size to 64. Plots of training and validation accuracy and loss are generated to visualize learning progress over epochs. These curves help detect overfitting, underfitting, or stable convergence. The techniques used has the following advantages, Batch Normalization stabilizes and accelerates training by normalizing layer inputs. Dropout prevents overfitting by randomly disabling neurons during training. Data Augmentation expands dataset variability, improving model generalization. Early Stopping stops training when there is no improvement in performance, avoiding overfitting. Model Checkpointing saves the best performing model weights during training. In this work, the facial features extracted using the Haar cascade technique are left eye, right eye, nose and mouth. The facial emotions are accurately predictable from the extracted features by the trained CNN model.

Table 3 shows the results of feature extraction. The features extracted are namely left eye, right eye, nose and mouth from the preprocessed facial images. The retrieved features are input into the CNN classifier, which is pre-trained for the categorization of seven facial expressions: happiness, sadness, disgust, surprise, fear, rage, and neutrality. From the features extracted the facial expressions are identified with utmost accuracy.

Table 4 illustrates the training and validation losses obtained at every five epochs. In machine learning, training loss and validation loss are the two metrics used to determine how efficiently a model fits training and test data, respectively. An epoch refers to the duration during which the entire training dataset has been processed by the network once. The ability of a model to fit the training data in each epoch is measured by its training loss. The validation loss is determined at the end of each epoch, which is the measure of how well a model fits new data. The gap between the training and validation loss curves shows whether the model is overfitting or underfitting. A large gap indicates overfitting. As training continues, both losses decrease, and between epochs 55 and 65 the gap becomes very small. This indicates that the model is well balanced and is neither overfitting nor underfitting. This state is achieved only when the input facial images both for training and testing the model are effectively feature extracted.

**Table 3:** Results of feature extraction

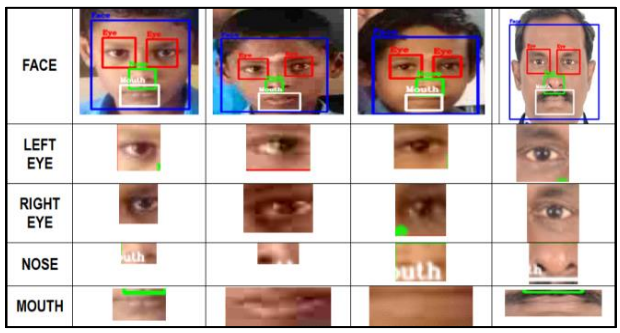


**Table 4:** Training and validation loss across epochs

| S. No | Epoch | Training Loss | Validation Loss | S. No | Epoch | Training Loss | Validation Loss |
|---|---|---|---|---|---|---|---|
| 1 | 5 | 1.940 | 1.515 | 9 | 45 | 1.017 | 1.010 |
| 2 | 10 | 1.607 | 1.323 | 10 | 50 | 0.979 | 0.982 |
| 3 | 15 | 1.406 | 1.195 | 11 | 55 | 0.865 | 0.865 |
| 4 | 20 | 1.269 | 1.148 | 12 | 60 | 0.846 | 0.846 |
| 5 | 25 | 1.186 | 1.117 | 13 | 65 | 0.824 | 0.824 |
| 6 | 30 | 1.127 | 1.091 | 14 | 70 | 0.806 | 0.854 |
| 7 | 35 | 1.084 | 1.063 | 15 | 75 | 0.772 | 0.883 |
| 8 | 40 | 1.048 | 1.032 | 16 | 80 | 0.750 | 0.915 |

Figure 4 shows that the result of extracting features from the preprocessed facial images is a reduce in both validation and training losses across epochs. The extent to which a model is overfitting or underfitting is significant from the difference between the training and validation loss curves. This demonstrates the contribution of the feature extraction method proposed in this research to find a solution for the overfitting condition. This is a reliable point of reference for future research work. The proposed model is precise and effective, which is evident from the disappearance of the discrepancy between the training and validation loss curves as the epoch reaches 50. There is still potential for refinement when striving for parity between training and validation errors.

An organized method for improving the model even further would be to search over the whole collection of potential hyperparameters for an optimal hyper-parameter. The learning rate, batch size, number of epochs, and other variables can be changed to achieve the optimal combination. One significant drawback in the search for the hyperparameter is the enormous demand on computer resources. In order to assist in this process of determining the best experiment or most promising sets of hyperparameters, TensorBoard's HParams dashboard offers a number of tools.

The training and validation accuracy of the model at the end of every five epochs is evident from Table 5. The performance of the model on the training dataset is the representation of the measure of training accuracy. It represents the ratio of accurately predicted instances to all instances in the training set. The degree to which the model generalizes to fresh, untested data is known as validation accuracy. A distinct set of data known as the validation dataset is used to determine the validation accuracy. The collections of facial images in the validation dataset are those that the model did not view during training. The above table clearly demonstrates an increase in both the training and validation accuracies as the epoch progresses. At the end of 80$^{th}$ epoch, the training accuracy reaches a maximum of 71.44 from 32.80, while the validation accuracy attains 68.14 from 42.10. At the 70$^{th}$ epoch, the validation accuracy reaches a maximum and then decreases with the epoch. It is apparent to stop the epochs between 70 and 75 in order to achieve utmost accuracy in both training and validation dataset. The feature extraction technique proposed in this work plays a vital role in increasing the accuracy of facial emotions prediction, which is observable from the experimental results obtained.

An increase in the accuracies of both the training and validation datasets as the number of epoch increases is noticeable from Figure 5. In the initial epoch, both training and validation accuracies are very low. Training the model can be ceased at that epoch value when accuracy reaches an optimum value. The experimental findings clearly show that the feature extraction method suggested in this work improves the accuracy rate of the model in predicting the seven basic facial expressions.

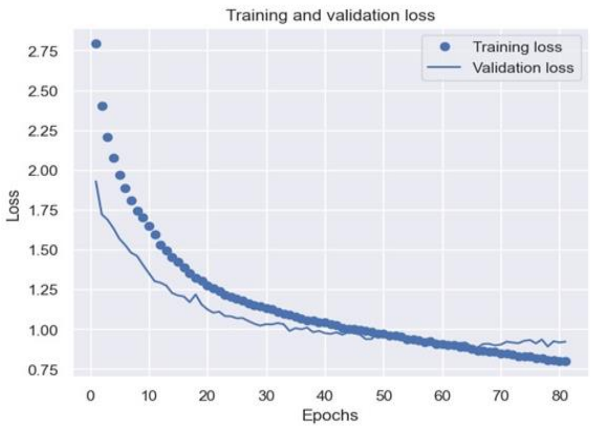


**Fig. 4:** Training and validation loss

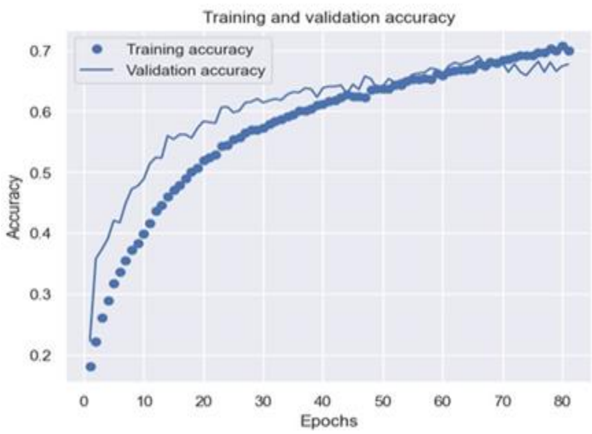


**Fig. 5:** Training and validation accuracy

**Table 5:** Training and validation accuracy across epochs

| S. no | Epoch | Training Accuracy | Validation Accuracy | S. no | Epoch | Training Accuracy | Validation Accuracy |
|---|---|---|---|---|---|---|---|
| 1 | 5 | 32.80 | 42.10 | 9 | 45 | 62.81 | 64.71 |
| 2 | 10 | 40.75 | 49.76 | 10 | 50 | 64.86 | 65.17 |
| 3 | 15 | 47.33 | 55.15 | 11 | 55 | 66.42 | 66.52 |
| 4 | 20 | 52.15 | 56.21 | 12 | 60 | 67.52 | 67.57 |
| 5 | 25 | 54.88 | 59.02 | 13 | 65 | 68.53 | 68.60 |
| 6 | 30 | 57.83 | 60.62 | 14 | 70 | 69.91 | 69.01 |
| 7 | 35 | 59.41 | 62.56 | 15 | 75 | 70.52 | 68.82 |
| 8 | 40 | 60.56 | 63.68 | 16 | 80 | 71.44 | 68.14 |

Table 6 illustrates the average training and validation accuracies across regular epoch intervals. It shows an increase in both the accuracies with the increase in number of epochs. The training and validation process of the facial expression recognition model stops between the 61st and 80th epochs once the accuracy becomes stable.

Figures 6 and 7 clearly show that both training and validation accuracy improve as the number of epochs increases. The four different epoch intervals are as follows, 1 to 20, 21 to 40, 41 to 60 and 61 to 80. The average training accuracy increases from 43.26% in the first epoch interval to 58.17% in the second epoch interval. Later further increases to 65.4% in the third epoch slot and finally reaches 70.1% in the last epoch interval. Similarly, the validation accuracy increases from 50.81 to 68.64% in the final epoch interval of 61 to 80. At the epoch interval of 61 to 80, both the training accuracy and validation accuracy of the facial expression recognition model show a noticeable improvement.

**Table 6:** Results of Accuracy across Epochs

| Classes | Epochs | Training Accuracy | Validation Accuracy |
|---|---|---|---|
| 1 | 1 to 20 | 43.26 | 50.81 |
| 2 | 21 to 40 | 58.17 | 61.47 |
| 3 | 41 to 60 | 65.40 | 65.99 |
| 4 | 61 to 80 | 70.10 | 68.64 |
| Average | | 59.23 | 61.72 |

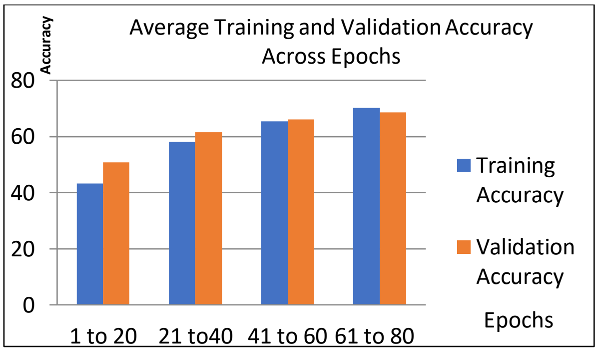


**Fig. 6:** Training and Validation Accuracy across Epoch intervals

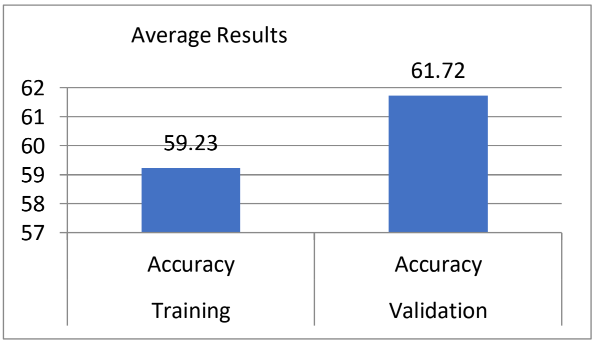


**Fig. 7:** Average of accuracies

## Conclusion

Facial expression recognition models usually predict seven basic human emotions: Happy, sad, angry, disgust, fear, surprise, and neutral. However, accurately identifying all these emotions is challenging. This research proposes a facial expression recognition system that uses facial images from the FER2013 dataset, covering all seven expressions, to train the model. The facial images are preprocessed from which the features are extracted using the suggested Haar Cascade algorithm and then the CNN model identifies seven fundamental facial expressions. The model's performance is evaluated using accuracy as the main metric. The results show a significant improvement in accuracy due to the use of the Haar Cascade technique for extracting important features from facial images. From the results, it implies that this method provides a simpler solution and achieves higher accuracy when compared to the traditional classifiers that make use of the same facial expression database with other feature extraction techniques. The combination of Haar Cascade and CNN is a well-established architecture. The innovation of this work is that it optimizes the pipeline for application in real-time classrooms. The proposed system achieves fast execution with low computational overhead, making it deployable on standard hardware without requiring specialized GPUs. This practical advantage ensures that the method can be scaled for real-world educational environments, providing a balance between accuracy, efficiency, and accessibility. Furthermore, training takes less time. Although the proposed approach demonstrates promising results, one limitation of this study is the relatively limited diversity of the test dataset. This constraint may restrict the generalizability of the findings across broader populations. Future work will focus on extending the evaluation to larger and more diverse datasets to enhance the robustness and applicability of the proposed system. In the future, this approach is intended to be evaluated in other databases, allowing for cross-database validation, as well as being used on wild facial images to predict facial expressions. The up gradation in the reliability of the model in future is acquired by using more appropriate combinations of feature extraction and feature selection techniques with deep learning models.

## Acknowledgment

We would like to thank Dwaraka Doss Goverdhan Doss Vaishnav College Chennai and University of Madras which offered moral support for accomplishing this research work.

## Funding Information

The authors received no financial support or funding to disclose.

DOI: 10.3844/jcssp.2025.2917.2927

## Author's Contributions

**Thambusamy Velumurugan:** Conceptualization, Methodology, and Supervision. He was responsible for the structural framework of the article, designing the study parameters, and overseeing the data collection process. He provided critical revisions for data analysis and gave final approval for the version to be published.

**Lakshminarayanan Meena:** Conceptualization, Investigation, and Formal Analysis. She originated the research idea, designed the data mining models, and executed the research plan. Additionally, she performed the software development and coding, conducted the implementation, and prepared the original draft of the manuscript.

## Ethics

The Research Article is original and has not been published anywhere. The corresponding author confirms that the other author has read and approved the manuscript and there are no ethical issues involved.

### *Conflict of Interest*

The authors have no conflicts of interest.

## References

Abouyahya, A., El Fkihi, S., Thami, R. O. H., & Aboutajdine, D. (2016). Features extraction for facial expressions recognition. *2016 5th International Conference on Multimedia Computing and Systems (ICMCS)*. Marrakech, Morocco. https://doi.org/10.1109/icmcs.2016.7905642

Choudhary, D., & Shukla, J. (2020). Feature Extraction and Feature Selection for Emotion Recognition using Facial Expression. *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*. New Delhi, India. https://doi.org/10.1109/bigmm50055.2020.00027

Dubuisson, S., Davoine, F., & Cocquerez, J.-P. (2001). *Automatic Facial Feature Extraction and Facial Expression Recognition*. 121–126. https://doi.org/10.1007/3-540-45344-x_19

Gogić, I., Manhart, M., Pandžić, I. S., & Ahlberg, J. (2020). Fast facial expression recognition using local binary features and shallow neural networks. *The Visual Computer*, *36*(1), 97–112. https://doi.org/10.1007/s00371-018-1585-8

González-Lozoya, S. M., de la Calleja, J., Pellegrin, L., Escalante, H. J., Medina, Ma. A., & Benitez-Ruiz, A. (2020). Recognition of facial expressions based on CNN features. *Multimedia Tools and Applications*, *79*(19–20), 13987–14007. https://doi.org/10.1007/s11042-020-08681-4

Gupta, S. K., Agrwal, S., Meena, Y. K., & Nain, N. (2011). A Hybrid Method of Feature Extraction for Facial Expression Recognition. *2011 Seventh International Conference on Signal Image Technology & Internet-Based Systems*. Internet-Based Systems (SITIS 2011), Dijon, France. https://doi.org/10.1109/sitis.2011.64

Happy, S. L., & Routray, A. (2015). Robust facial expression classification using shape and appearance features. *2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR)*. Kolkata, India. https://doi.org/10.1109/icapr.2015.7050661

Huang, C.-L., & Huang, Y.-M. (1997). Facial Expression Recognition Using Model-Based Feature Extraction and Action Parameters Classification. *Journal of Visual Communication and Image Representation*, *8*(3), 278–290. https://doi.org/10.1006/jvci.1997.0359

Ilbeygi, M., & Shah-Hosseini, H. (2012). A novel fuzzy facial expression recognition system based on facial feature extraction from color face images. *Engineering Applications of Artificial Intelligence*, *25*(1), 130–146. https://doi.org/10.1016/j.engappai.2011.07.004

Janu, N., Mathur, P., Gupta, S. K., & Agrwal, S. L. (2017). Performance analysis of frequency domain based feature extraction techniques for facial expression recognition. *2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence*. Noida, India. https://doi.org/10.1109/confluence.2017.7943220

Kola, D. G. R., & Samayamantula, S. K. (2021). A novel approach for facial expression recognition using local binary pattern with adaptive window. *Multimedia Tools and Applications*, *80*(2), 2243–2262. https://doi.org/10.1007/s11042-020-09663-2

Kommineni, J., Mandala, S., Sunar, M. S., & Chakravarthy, P. M. (2021). Accurate computing of facial expression recognition using a hybrid feature extraction technique. *The Journal of Supercomputing*, *77*(5), 5019–5044. https://doi.org/10.1007/s11227-020-03468-8

Krithika, L. B., & Priya, G. G. L. (2021). Graph based feature extraction and hybrid classification approach for facial expression recognition. *Journal of Ambient Intelligence and Humanized Computing*, *12*(2), 2131–2147. https://doi.org/10.1007/s12652-020-02311-5

Kumar, S., Bhuyan, M. K., & Chakraborty, B. K. (2016). Extraction of informative regions of a face for facial expression recognition. *IET Computer Vision*, *10*(6), 567–576. https://doi.org/10.1049/iet-cvi.2015.0273

Lai, C.-C., & Ko, C.-H. (2014). Facial expression recognition based on two-stage features extraction. *Optik*, *125*(22), 6678–6680. https://doi.org/10.1016/j.ijleo.2014.08.052

Li, B., & Lima, D. (2021). Facial expression recognition via ResNet-50. *International Journal of Cognitive Computing in Engineering*, *2*, 57–64. https://doi.org/10.1016/j.ijcce.2021.02.002

Li, J., Jin, K., Zhou, D., Kubota, N., & Ju, Z. (2020). Attention Mechanism-Based CNN for Facial Expression Recognition. *Neurocomputing*, *411*, 340–350. https://doi.org/10.1016/j.neucom.2020.06.014

Li, Y., Lu, G., Li, J., Zhang, Z., & Zhang, D. (2023). Facial Expression Recognition in the Wild Using Multi-Level Features and Attention Mechanisms. *IEEE Transactions on Affective Computing*, *14*(1), 451–462. https://doi.org/10.1109/taffc.2020.3031602

Liu, C., Hirota, K., Ma, J., Jia, Z., & Dai, Y. (2021). Facial Expression Recognition Using Hybrid Features of Pixel and Geometry. *IEEE Access*, *9*, 18876–18889. https://doi.org/10.1109/access.2021.3054332

Luo, Y., Wu, C., & Zhang, Y. (2013). Facial expression feature extraction using hybrid PCA and LBP. *The Journal of China Universities of Posts and Telecommunications*, *20*(2), 120–124. https://doi.org/10.1016/s1005-8885(13)60038-2

Mattela, G., & Gupta, S. K. (2018). Facial Expression Recognition Using Gabor-Mean-DWT Feature Extraction Technique. *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*. Noida. https://doi.org/10.1109/spin.2018.8474206

Mayya, V., Pai, R. M., & Manohara Pai, M. M. (2016). Automatic Facial Expression Recognition Using DCNN. *Procedia Computer Science*, *93*, 453–461. https://doi.org/10.1016/j.procs.2016.07.233

Meena, L., & Velmurugan, T. (2023). Optimizing Facial Expression Recognition through Effective Preprocessing Techniques. *Journal of Computer and Communications*, *11*(12), 86–101. https://doi.org/10.4236/jcc.2023.1112006

Nan, Y., Ju, J., Hua, Q., Zhang, H., & Wang, B. (2022). A-MobileNet: An approach of facial expression recognition. In *Alexandria Engineering Journal* (Vol. 61, Issue 6, pp. 4435–4444). https://doi.org/10.1016/j.aej.2021.09.066

Pei, J., & Shan, P. (2019). A Micro-expression Recognition Algorithm for Students in Classroom Learning Based on Convolutional Neural Network. *Traitement Du Signal*, *36*(6), 557–563. https://doi.org/10.18280/ts.360611

Rizwan, S. A., Jalal, A., & Kim, K. (2020). An Accurate Facial Expression Detector using Multi-Landmarks Selection and Local Transform Features. *2020 3rd International Conference on Advancements in Computational Sciences (ICACS)*. Lahore, Pakistan. https://doi.org/10.1109/icacs47775.2020.9055954

Sadeghi, H., & Raie, A.-A. (2019). Human vision inspired feature extraction for facial expression recognition. *Multimedia Tools and Applications*, *78*(21), 30335–30353. https://doi.org/10.1007/s11042-019-07863-z

Siddiqi, M. H., Farooq, F., & Lee, S. (2012). A robust feature extraction method for human facial expressions recognition systems. *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*. IVCNZ ’12: Image and Vision Computing New Zealand, Dunedin New Zealand. https://doi.org/10.1145/2425836.2425924

Singh, S., & Nasoz, F. (2020). Facial Expression Recognition with Convolutional Neural Networks. *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. Las Vegas, NV, USA. https://doi.org/10.1109/ccwc47524.2020.9031283

Tsai, H.-H., & Chang, Y.-C. (2018). Facial expression recognition using a combination of multiple facial features and support vector machine. *Soft Computing*, *22*(13), 4389–4405. https://doi.org/10.1007/s00500-017-2634-3

Tseng, C.-H., & Chen, Y.-H. (2018). A camera-based attention level assessment tool designed for classroom usage. *The Journal of Supercomputing*, *74*(11), 5889–5902. https://doi.org/10.1007/s11227-017-2122-7

Vedantham, R., & Reddy, E. S. (2020). A robust feature extraction with optimized DBN-SMO for facial expression recognition. *Multimedia Tools and Applications*, *79*(29–30), 21487–21512. https://doi.org/10.1007/s11042-020-08901-x

Wang, H., Wei, S., & Fang, B. (2020). Facial expression recognition using iterative fusion of MO-HOG and deep features. *The Journal of Supercomputing*, *76*(5), 3211–3221. https://doi.org/10.1007/s11227-018-2554-8

Zhao, X., & Zhang, S. (2016). A Review on Facial Expression Recognition: Feature Extraction and Classification. *IETE Technical Review*, *33*(5), 505–517. https://doi.org/10.1080/02564602.2015.1117403

Zhang, L., Tjondronegoro, D., & Chandran, V. (2012). Discovering the Best Feature Extraction and Selection Algorithms for Spontaneous Facial Expression Recognition. *2012 IEEE International Conference on Multimedia and Expo (ICME)*. Melbourne, Australia. https://doi.org/10.1109/icme.2012.97

Zhou, G., Zhan, Y., & Zhang, J. (2006). Facial Expression Recognition Based on Selective Feature Extraction. *Sixth International Conference on Intelligent Systems Design and Applications*. China. https://doi.org/10.1109/isda.2006.253872